UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/631,174	08/02/2000	Lawrence D. K. B. Dwyer	10001219-1	7798

7590 08/10/2004

Hewlett-Packard Company
Intellectual Property Administration
P O Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	10

DATE MAILED: 08/10/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/631,174	Applicant(s) DWYER ET AL.	
	Examiner Aimee J Li	Art Unit 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 June 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-21 and new claims 22-30 have been considered. Claims 20 and 21 have been amended as per Applicant's request. New claims 22-30 have been added as per Applicant's request.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Novak et al., U.S. Patent Number 5,809,522 (herein referred to as Novak) in view of Gulsen et al., U.S. Patent Number 5,727,211 (herein referred to as Gulsen)

4. Referring to claim 1, Novak has taught a computer system for efficiently executing instructions of computer programs, comprising:

a. Processing circuitry having a pipeline, said pipeline configured to execute instructions from one of a plurality of programs, said processing circuitry further configured to stop executing said one program during a first context switch in response to a first context switch command and to resume executing said one program during a second context switch in response to a second context switch command (Novak Abstract; column 4, lines 56-57; column 8, lines 47-65; Figure 2; and Figure 5). In regards to Novak, it is inherent that the system is pipelined,

since the system being described is an x86 microprocessor system. Please see the provided InstantWeb Online Computing Dictionary attachments.

b. Cache memory (Novak Abstract; column 1, lines 12-17; and Figure 1);

c. Computer memory having a plurality of addresses (Novak Abstract; column 1, lines 12-17; columns 2-3, lines 61-16; and Figure 1).

5. Novak has not explicitly taught memory control circuitry coupled to said processing circuitry, said memory control circuitry, in response to said second context switch command, configured to identify one of said addresses of said computer memory that is storing a data value previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch, said memory control circuitry further configured to retrieve said data value from said computer memory in response to said second context switch command and to store said retrieved data value in said cache memory. However, Novak has taught identifying data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch (Novak Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5), but has not explicitly taught how to handle when the second task must overwrite data for the first task in shared resources. Gulsen has explicitly taught memory control circuitry coupled to said processing circuitry, said memory control circuitry, in response to said second context switch command, configured to identify one of said addresses of said computer memory that is storing a data value previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch, said memory control circuitry further configured to retrieve said data value from said computer memory in response to said second

context switch command and to store said retrieved data value in said cache memory (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory management routine of Gulsen reduces the time necessary to restore the current task and preserves the data of the current task for future use by only copying and restoring information that would be written over by the next task in shared resources. Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the memory management of Gulsen in the device of Novak to reduce the time necessary to restore the current task and preserve data needed for future use.

6. Referring to claim 2, Novak has taught wherein said processing circuitry is further configured to execute instructions of another of said computer programs in response to said first context switch command (Novak Abstract; column 8, lines 47-65; Figure 2; and Figure 5).

7. Referring to claim 3, Novak has taught wherein said memory control circuitry is further configured to determine, in response to said second context switch command, whether said data value was utilized by said processing circuitry to execute an instruction within a specified time period prior to said first context switch (Novak Abstract; column 3, lines 38-52; column 8, lines 13-65; Figure 2; and Figure 5).

8. Referring to claim 4, Novak has taught wherein said memory control circuitry is configured to maintain a plurality of mappings, each of said mappings respectively correlating a data value stored in said cache memory with one of said memory addresses of said computer memory (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5),

said memory control circuitry further configured to maintain a bit of information that is associated with one of said mappings, said memory control circuitry configured to assert said bit when a data value correlated with a computer memory address via said one mapping is utilized to execute an instruction of said one program, said memory control circuitry further configured to deassert said bit periodically (Novak Abstract; column 3, lines 38-51; column 8, lines 13-65; Figure 2; and Figure 5).

9. Referring to claim 5, Novak has taught wherein said memory control circuitry is further configured to determine, in response to said second context switch command arid based on said bit, whether said data value was recently utilized by said processing circuitry to execute an instruction prior to said first context switch (Novak Abstract; column 8, lines 13-65; Figure 2; and Figure 5).

10. Referring to claim 6, Novak has taught wherein said memory control circuitry is further configured to store said mappings and said bit to said computer memory in response to said first context switch command and to retrieve said mappings and said bit from said computer memory in response to said second context switch command (Novak Abstract; column 3, lines 38-52; column 8, lines 13-65; Figure 2; and Figure 5).

11. Referring to claim 7, Novak has taught a computer system for efficiently executing instructions of computer programs, comprising:

a. Processing circuitry having a pipeline, said pipeline configured to execute instructions from one of a plurality of programs, said processing circuitry further configured to stop executing said one program during a first context switch in response to a first context switch command and to resume executing said one

program during a second context switch in response to a second context switch command (Novak Abstract; column 8, lines 47-65; Figure 2; and Figure 5). In regards to Novak, it is inherent that the system is pipelined, since the system being described is an x86 microprocessor system. Please see the provided InstantWeb Online Computing Dictionary attachments.

b. Cache memory (Novak Abstract; column 1, lines 12-17; and Figure 1);

c. Computer memory having a plurality of addresses (Novak Abstract; column 1, lines 12-17; columns 2-3, lines 61-16; and Figure 1); and

12. Novak has not explicitly taught memory control circuitry coupled to said processing circuitry, said memory control circuitry configured to maintain a plurality of mappings, said mappings respectively correlating data values previously written by said pipeline execution of an instruction and stored in said cache memory with said memory addresses of said computer memory, said memory control circuitry configured to store said mappings in said computer memory in response to said first context switch command and to retrieve said data values from said addresses that are identified by said mappings stored in said computer memory in response to said second context switch command, said memory control circuitry further configured to store in said cache memory said retrieved data values. However, Novak has taught identifying data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch (Novak Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5), but has not explicitly taught how to handle when the second task must overwrite data for the first task in shared resources. Gulsen has explicitly taught memory control circuitry coupled to said processing

circuitry, said memory control circuitry configured to maintain a plurality of mappings, said mappings respectively correlating data values previously written by said pipeline execution of an instruction and stored in said cache memory with said memory addresses of said computer memory, said memory control circuitry configured to store said mappings in said computer memory in response to said first context switch command and to retrieve said data values from said addresses that are identified by said mappings stored in said computer memory in response to said second context switch command, said memory control circuitry further configured to store in said cache memory said retrieved data values (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory management routine of Gulsen reduces the time necessary to restore the current task and preserves the data of the current task for future use by only copying and restoring information that would be written over by the next task in shared resources. Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the memory management of Gulsen in the device of Novak to reduce the time necessary to restore the current task and preserve data needed for future use.

13. Referring to claim 8, Novak has taught wherein said processing circuitry is further configured to execute instructions of another of said computer programs in response to said first context switch command (Novak Abstract; column 8, lines 47-65; Figure 2; and Figure 5).

14. Referring to claim 9, Novak has taught wherein said memory control circuitry is further configured to maintain utilization data indicative of which of said memory addresses are storing

data values accessed within a specified time period prior to said first context switch (Novak Abstract; column 3, lines 38-52; column 8, lines 13-65; Figure 2; and Figure 5), and wherein said memory control circuitry, based on said mappings and said utilization data, is further configured to select for retrieval data values identified by one of said mappings and accessed within said specified time period (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5), wherein each of said retrieved data values is a data value selected by said memory control circuitry based on said utilization data (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

15. Referring to claim 10, Novak has taught wherein said memory control circuitry is farther configured to store said utilization data in said computer memory in response to said first context switch command and to retrieve said utilization data and said mappings in response to said second context switch command (Novak Abstract; column 3, lines 38-52; column 8, lines 13-65; Figure 2; and Figure 5).

16. Referring to claim 11, Novak has taught wherein said utilization data is a plurality of bits respectively associated with said mappings, wherein said memory control circuitry, for each data value accessed by said memory control circuitry, is configured to assert the bit associated with the mapping that correlates said each data value with one of said computer memory addresses, and wherein said memory control circuitry is configured to periodically deassert each of said plurality of bits (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

17. Referring to claim 12, Novak has taught a method for efficiently executing instructions of computer programs, comprising the steps of:

a. Executing a plurality of computer programs in an interleaved fashion; switching which of said computer programs is being executed in said executing step (Novak Abstract; column 8, lines 47-65; and Figure 5);

b. Retrieving said data value from said cache memory in response to said executing step (Novak Abstract; column 1, lines 12-17; columns 2-3, lines 61-38; and Figure 1).

18. Novak has not explicitly taught

a. Storing, prior to said switching step, at an address in computer memory a data value previously written by a pipeline in execution of an instruction corresponding to one of said computer programs in said executing step; identifying said address in response to said switching step;

b. Retrieving said data value from said address based on said identifying step and in response to said switching step;

c. Storing said retrieved data value in cache memory.

19. However, Novak has taught identifying data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch (Novak Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5), but has not explicitly taught how to handle when the second task must overwrite data for the first task in shared resources. In regards to Novak, it is inherent that the system is pipelined, since the system being described is an x86 microprocessor system. Please see the provided InstantWeb Online Computing Dictionary attachments. Gulsen has explicitly taught

a. Storing, prior to said switching step, at an address in computer memory a data value previously written by a pipeline in execution of an instruction corresponding to one of said computer programs in said executing step; identifying said address in response to said switching step (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6);

b. Retrieving said data value from said address based on said identifying step and in response to said switching step (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6);

c. Storing said retrieved data value in cache memory (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6).

20. A person of ordinary skill in the art at the time the invention was made would have recognized that the memory management routine of Gulsen reduces the time necessary to restore the current task and preserves the data of the current task for future use by only copying and restoring information that would be written over by the next task in shared resources. Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the memory management of Gulsen in the device of Novak to reduce the time necessary to restore the current task and preserve data needed for future use.

21. Referring to claim 13, Novak has taught wherein said executing step further includes the step of executing instructions of a computer program in response to said switching step, and wherein said method further comprises the steps of:

a. Determining that said address is storing a data value previously utilized in said executing step to execute an instruction of said computer program (Novak Abstract and column 3, lines 38-52); and

b. Performing said identifying step based on said determining step (Novak Abstract; column 8, lines 13-65; Figure 2; and Figure 5).

22. Referring to claim 14, Novak has taught further comprising the steps of:

a. Correlating, respectively, data values stored in said cache memory with addresses of said computer memory (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5);

b. Asserting a bit each time a data value correlated with said address identified in said identifying step is accessed in response to said executing step (Novak Abstract; column 3, lines 38-51; column 8, lines 13-65; Figure 2; and Figure 5); and

c. Periodically deasserting said bit (Novak Abstract; column 3, lines 38-51; column 8, lines 13-65; Figure 2; and Figure 5).

23. Referring to claim 15, Novak has taught wherein said executing step further includes the step of executing instructions of a computer program in response to said switching step, and wherein said method further comprises the steps of:

a. Determining, based on said bit, that said address identified in said identifying step is storing a data value previously utilized in said executing step-to execute an instruction of said computer program (Novak Abstract; column 8, lines 13-65; Figure 2; and Figure 5); and

b. Performing said identifying step based on said determining step (Novak Abstract; column 8, lines 13-65; Figure 2; and Figure 5).

24. Referring to claim 16, Novak has taught a method for efficiently executing instructions of computer programs, comprising the steps of:

a. Executing instructions from a computer program (Novak Abstract; column 8, lines 13-65; Figure 2; and Figure 5);

b. Halting said executing step during a first context switch in response to a first context switch command (Novak Abstract; column 8, lines 47-65; Figure 2; and Figure 5);

c. Resuming said executing step during a second context switch in response to a second context switch command; maintaining a plurality of mappings (Novak Abstract; column 3, lines 38-51; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5);

25. Novak has not explicitly taught

a. Correlating, via said mappings, data values previously written by a pipeline during the executing step and stored in a cache memory with memory addresses of computer memory outside of said cache memory; storing said mappings in said computer memory in response to said first context switch command;

b. Retrieving, based on said mappings and in response to said second context switch command, at least one data value from at least one of said addresses identified by said mappings; and storing said at least one retrieved data value in said cache memory.

26. However, Novak has taught identifying data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch (Novak Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5), but has not explicitly taught how to handle when the second task must overwrite data for the first task in shared resources. In regards to Novak, it is inherent that the system is pipelined, since the system being described is an x86 microprocessor system. Please see the provided InstantWeb Online Computing Dictionary attachments. Gulsen has explicitly taught

a. Correlating, via said mappings, data values previously written by a pipeline during the executing step and stored in a cache memory with memory addresses of computer memory outside of said cache memory; storing said mappings in said computer memory in response to said first context switch command (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6);

b. Retrieving, based on said mappings and in response to said second context switch command, at least one data value from at least one of said addresses identified by said mappings; and storing said at least one retrieved data value in said cache

memory (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6).

27. A person of ordinary skill in the art at the time the invention was made would have recognized that the memory management routine of Gulsen reduces the time necessary to restore the current task and preserves the data of the current task for future use by only copying and restoring information that would be written over by the next task in shared resources. Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the memory management of Gulsen in the device of Novak to reduce the time necessary to restore the current task and preserve data needed for future use.

28. Referring to claim 17, Novak has taught further comprising the steps of:

a. Maintaining utilization data indicative of which of said memory addresses are storing data values accessed within a specified time period prior to said first context switch (Novak Abstract; column 3, lines 38-52; column 8, lines 13-65; Figure 2; and Figure 5); and

b. Selecting, based on said mappings and said utilization data, data values accessed within said specified time period (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5),

c. Wherein said retrieving step includes the step of retrieving each data value selected in said selecting step (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

29. Referring to claim 18, Novak has taught further comprising the steps of:

a. Storing said utilization data in said computer memory in response to said first context switch command (Novak Abstract; column 3, lines 38-52; column 8, lines 13-54; Figure 2; and Figure 5); and

b. Retrieving said utilization data and said mappings in response to said second context switch command (Novak Abstract; column 3, lines 38-52; column 8, lines 13-54; Figure 2; and Figure 5).

30. Referring to claim 19, Novak has taught wherein said utilization data is a plurality of bits respectively associated with said mappings, and wherein said method further comprises the steps of:

a. Asserting each of said bits associated respectively with each of said mappings that identifies a data value accessed in response to said executing step (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5); and

b. Periodically deasserting each of said bits (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

31. Referring to claim 20, Novak has taught a computer system for efficiently executing instructions of computer programs, comprising:

a. A processing unit (Novak Abstract; column 4, lines 56-57; column 8, lines 47-65; Figure 2; and Figure 5);

b. Computer memory residing outside of the processing unit (Novak Abstract; column 1, lines 12-17; columns 2-3, lines 61-16; and Figure 1);

c. Cache memory (Novak Abstract; column 1, lines 12-17; and Figure 1); and

d. Logic configured to store in said computer memory a value indicative of cache memory usage and a mapping associated with said value (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5),

e. Said processing unit continuing execution of said first process with the retrieved data when the processing unit context switches out the second process and context switches in the first process (Novak Abstract; column 4, lines 56-57; column 8, lines 47-65; Figure 2; and Figure 5).

32. Novak has not explicitly taught said mapping indicative of a location in said computer memory storing data previously requested or previously written by an instruction of a first process being executed by the processing unit when the processing unit context switches out the first process for processing of a second process, the logic further configured to retrieve said data, based on said value, and store said data in said cache. However, Novak has taught identifying data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch (Novak Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5), but has not explicitly taught how to handle when the second task must overwrite data for the first task in shared resources. Gulsen has explicitly taught said mapping indicative of a location in said computer memory storing data previously requested or previously written by an instruction of a first process being executed by the processing unit when the processing unit context switches out the first process for processing of a second process, the logic further configured to retrieve said data, based on said value, and store said data in said cache (Gulsen Abstract; column 2, line 51 to

column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory management routine of Gulsen reduces the time necessary to restore the current task and preserves the data of the current task for future use by only copying and restoring information that would be written over by the next task in shared resources. Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the memory management of Gulsen in the device of Novak to reduce the time necessary to restore the current task and preserve data needed for future use.

33. Referring to claim 21, Novak has taught a method for efficiently executing instructions of computer programs, comprising the steps of:

a. Storing in memory outside of a processing unit (Novak Abstract; column 1, lines 12-17; columns 2-3, lines 61-16; and Figure 1) a value indicative of cache memory usage (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5);

b. Storing in said memory a mapping corresponding to said value indicative of cache memory usage (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5), said mapping indicative of a location in computer memory storing data previously requested or previously written by an instruction of a first process being executed by the processing unit when the processing unit context switches out the first process for processing of a second process (Novak

Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5); and

c. Continuing execution of said first process with the data retrieved in the retrieving step (Novak Abstract; column 4, lines 56-57; column 8, lines 47-65; Figure 2; and Figure 5).

34. Novak has not explicitly taught retrieving said data, based upon said value, when the processing unit context switches out the second process and context switches in the first process. However, Novak has taught identifying data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch (Novak Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5), but has not explicitly taught how to handle when the second task must overwrite data for the first task in shared resources. Gulsen has explicitly taught retrieving said data, based upon said value, when the processing unit context switches out the second process and context switches in the first process (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory management routine of Gulsen reduces the time necessary to restore the current task and preserves the data of the current task for future use by only copying and restoring information that would be written over by the next task in shared resources. Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the memory management of Gulsen in

the device of Novak to reduce the time necessary to restore the current task and preserve data needed for future use.

35. Referring to claims 22 and 25, Novak has taught wherein said cache memory comprises a cache line (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

36. Referring to claims 23 and 26, Novak has taught wherein said value is indicative of whether said processing unit has accessed said cache line during a particular time period (Novak Abstract; column 3, lines 38-52; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

37. Referring to claim 24, Novak has taught wherein said value indicative of said cache memory usage is defined by a flag, said logic configured to assert said flag when said first process uses said cache line (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

38. Referring to claim 27, Novak has taught the step of asserting a flag when said first process uses said cache line (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

39. Referring to claim 28, Novak has taught wherein said memory control circuitry is further configured to track usage frequency of cache lines in said cache memory during execution of said program (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

40. Referring to claim 29, Novak has taught wherein said memory control circuitry is configured to identify said addresses of said computer memory and to retrieve said data based

upon said tracked usage frequency of said cache lines (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5).

41. Referring to claim 30, Novak has taught a system, comprising:

a. A processing unit(Novak Abstract; column 4, lines 56-57; column 8, lines 47-65; Figure 2; and Figure 5);

b. Cache memory comprising a cache line, the cache line comprising data used by a first process during execution by said processing unit (Novak Abstract; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5); and

c. Logic configured to track an usage frequency of said cache line (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5), said logic further configured to store in computer memory a value indicative of the usage frequency (Novak Abstract; column 3, lines 38-52; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 2; and Figure 5) and a mapping associated with said data used from said cache line by said first process upon a first context switch (Novak Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5).

42. Novak has not explicitly taught said logic further configured to preload said data into said cache upon a second context switch based on said value. However, Novak has taught identifying data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch (Novak Abstract; columns 3-4, lines 56-4; column 4, lines 57-65; column 8, lines 13-65; Figure 1; Figure 2; and Figure 5), but has not explicitly

taught how to handle when the second task must overwrite data for the first task in shared resources. Gulsen has explicitly taught said logic further configured to preload said data into said cache upon a second context switch based on said value (Gulsen Abstract; column 2, line 51 to column 3, line 15; column 3, lines 24-47; column 3, lines 23-25; column 5, lines 26-42; column 11, line 40 to column 12, line 31; Figure 1; and Figure 6). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory management routine of Gulsen reduces the time necessary to restore the current task and preserves the data of the current task for future use by only copying and restoring information that would be written over by the next task in shared resources. Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the memory management of Gulsen in the device of Novak to reduce the time necessary to restore the current task and preserve data needed for future use.

Response to Arguments

43. Applicant's arguments with respect to claims 20-30 have been considered but are moot in view of the new ground(s) of rejection.

44. Applicant's arguments filed 10 June 2004 have been fully considered but they are not persuasive.

45. Applicant's argue in essence on pages 14-16

> ...there is nothing **in the cited art** to indicate that it would have been obvious to one of ordinary skill in the art, at the time of filing of the instant application, to incorporate the memory management system of *Gulsen* to reduce the time necessary to restore the current task and preserve data needed for future use.

46. This has not been found persuasive. In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves **or in the knowledge generally available to one of ordinary skill in the art**. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the cited art does not have to provide a motivation to combine to references. It must only have been obvious to a person of ordinary skill in the art at the time the invention was made.

Conclusion

47. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (703) 305-7596. The examiner can normally be reached on M-T 7:30am-5:00pm.

48. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

49. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
August 9, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100